User-Centered Planning

A Discussion on Automated Planning in the Presence of Human Users

Pascal Bercher, Daniel Höller, Gregor Behnke, Susanne Biundo

Institute of Artificial Intelligence, Ulm University, Germany

forename.surname@uni-ulm.de

Abstract

We discuss how the Hybrid Planning paradigm can be exploited when planning in the presence of human users:

- Intuitive plan generation process (useful for incorporating users into the process, called Mixed Initiative Planning).
- Plan execution, which includes:
  - Plan linearization,
  - Presentation of plans,
  - Monitoring of executed steps.
- Plan repair.
- Plan explanation.

### Example Application: Assembling a Home Entertainment System

To exemplify the discussed user-centered planning capabilities, we use a running example:

A user wants to assemble his home entertainment system. The home entertainment system’s subdevices must be connected with each other using the correct cables and adapters.

We implemented an assistance system that implements various of the desired user-centered planning capabilities:

- Its instructions show in detail how to set up the theater.
- The purpose of any presented instruction can be explained.
- The system can cope with execution errors (broken cables).

### Definition (Plan)

A plan \( P \) is a partially ordered sequence of tasks on \( P = (T, \preceq, M) \), where:

- \( T \), \( \preceq \), and \( M \) are finite sets of abstract and primitive tasks, respectively.
- Each task \( t \in T \) is a task with a unique label \( \ell(t) \).
- A plan \( P \) is a pair \( (P, \ell) \) consisting of a conjunction of literals over the set of state variables.

A hybrid planning problem is a tuple \( H = (G, \phi, \Omega, g, p) \), where:

- \( G \) is the planning domain.
- \( \phi \) and \( g \) are the initial and the goal state, respectively.
- \( p \) is the initial plan. As usual in PDDL planning, it contains two special actions that encode \( \phi \) and \( g \).

### Problems of Plan Repair

Plan repair serves the purpose of repairing partial plans in the following cases:

- The repaired plan is still a solution to the original problem.
- All executed steps are contained in a new solution (marked as executed), so one does not have to start over.
- Issues of plan repair.
- When to repair? Also when "short cuts" are possible?
- How to repair? Plan stability/similarity vs. optimization criteria.

### Plan Explanation

To obtain full transparency of a system, it needs be able to justify its behavior/decisions. To that purpose we can explain certain properties of plans, e.g.:

- Why has action \( A \) been performed before it?
- Why does action \( C \) manipulate object \( ? \)?
- Why is action \( ? \) part of the plan, anyway?

These questions can be answered in natural language. This is done based on a proof in an automatic system. That system formalizes:

- The decomposition hierarchy.
- The solution plan’s causal structure (causal links).

### Plan Execution

The plan execution component monitors the execution state of the plan: in case the current state deviates from the anticipated one, plan repair is initiated.

Plan Explanation (Example)

In the plan depicted above, the user wants to know why he should perform action plugIn(BR,CINCH,AUDIO).

The analysis of the causal structure (highlighted in the depicted plan) corresponds to a sequence of single proof steps:

1. \( \lnot \text{plugIn(BR,CINCH,AUDIO)} \)
2. \( \lnot \text{plugIn(BR,CINCH,AUDIO)} \)
3. \( \lnot \text{plugIn(CINCH,AV-Rec,AUDIO)} \)
4. \( \text{plugIn(CINCH,AV-Rec,AUDIO)} \)
5. \( \text{goal-task-for-g} \)
6. \( \text{goal-task-for-g} \)

These proof steps can be translated into natural language:

- You have to connect the Blu-ray player with the CINCH cable to be able to connect that cable with the AV receiver, so the audio signal is transmitted from the Blu-ray player to the AV receiver.

### Plan Presentation

After an execution sequence has been selected, the actions have to be communicated to the user (assuming they are not automatically executed by a system).

Consider the action plugIn(BR,CINCH,AUDIO).

Given that the knowledge base stores appropriate pictures and videos for the objects used by the action, detailed interfaces can be generated automatically. Using templates, the instructions can also be presented using natural language.

### Plan Generation Process

How are solutions generated and why and how is this process beneficial for planning with or for humans?

By step-wise refining the initial plan into a solution plan, a user can be smoothly integrated into the plan generation process.

- Hierarchical Refinement:
  - Abstract tasks are step-wise refined into more primitive courses of action – similar to human problem solving.
  - Abstract tasks show preconditions and effects, which gives them a clear semantics and allows to generate plans on an abstract level.
- Goal-Directed Refinement:
  - Missing steps are inserted to analyze causal dependencies
  - Causal dependencies are explicitly represented via causal links. That way “unfinished” parts of the plan can be identified.

### Definition (Plan Generation Domain and Problem)

A hybrid planning problem is a tuple \( H = (G, \phi, \Omega, g, p) \), where:

- \( G \) is the planning domain.
- \( \phi \) and \( g \) are the initial and the goal state, respectively.
- \( p \) is the initial plan. As usual in PDDL planning, it contains two special actions that encode \( \phi \) and \( g \).

### Definition (Solution Plan)

A plan \( P \) is a solution if:

- \( P \) is a refinement of \( \Pi \), i.e., \( P \) can be obtained from \( \Pi \) via:
  - Decomposition: given a plan \( \Pi = (T, \preceq, M) \), use method \( \Pi' \subseteq M \) to replace \( I \ precedes \ P \) by \( P \). Causal links and orderings are inherited.
  - Insertion of ordering constraints.
  - Insertion of tasks. This refinement is optional.
  - Insertion of causal links.
- Every linearization of \( P \) is executable in \( t \), and satisfies \( g \).

In the absence of causal links, the respective problem classes are called HTN planning or HTN planning with task inheritance, depending on whether the insertion of tasks is allowed.

### Mixed Initiative Planning

Plan repair.

In the absence of causal links, the respective problem classes are called HTN planning or HTN planning with task inheritance, depending on whether the insertion of tasks is allowed.